PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES
(Appeal Brief-2)

APPLICANT(S): Vandersluis. EXAMINER: Nguyen, Cindy

SERIAL NO.: 09/900,079 ART GROUP: 2161

FILED July 6, 2001 Case No.: XAW-0102

ENTITLED: System and Method for Converting Data in a First Hierarchical Data Scheme into a Second Hierarchical Data Scheme

I hereby certify that this correspondence is being deposited with the United States Postal Service as first-class mail in an envelope addressed to: Honorable Commissioner of Patents and Trademarks, P.O. Box 1450, Alexandria, VA 22313-1450 on:

_12/13/07_

Date of Deposit

Dale B. Halling

Attorney of Record

Signature

Rev. 6/93

Honorable Commissioner of

Patents and Trademarks

P.O. Box 1450

Alexandria, VA 22313-1450

Sir:

This is an appeal from the rejection of claims 1-29 of the Office Action dated July 27, 2007, which was a reopened prosecution in view of the Appeal Brief filed on March 26, 2007, which was a reopened prosecution in view of the Appeal Brief filed on August 26, 2005,. A Supplemental Notice of Appeal was filed on October 24, 2007. The original Appeal was from the rejection of claims 1-29 of the Office Action dated March 23, 2005. This application was filed on July 6, 2001. Appellant submits this Appeal Brief pursuant to 35 U.S.C. §134 and 37 C.F.R. § 1.191 in furtherance of the Notice of Appeal filed in this case on December 22, 2006. The fees required under 37 C.F.R. §1.17(b) and 37 C.F.R. § 41.20(2) and any other necessary fees were paid with respect to the original Appeal Brief filed on August 26, 2005 according to MPEP 1204.01 no additional fees are due.

I. **Real Party In Interest**

The real party in interest is: XAware Inc., a Colorado corporation, having a place of business at 5555 Tech Center Dr., Suite 200, Colorado Springs, CO 80919. See the Assignment recorded at Reel 012236, Frame 0990.

II. **Related Appeals And Interferences**

The Original Appeal was from the rejection of claims 1-29 of the Office Action dated March 23, 2005. There have been two subsequent appeals, including the present appeal. There are no interferences related to the present appeal.

III. **Status Of Claims**

Claims 1-29 (see Appendix) are pending in this application. Claims 1-29 are rejected and are involved in this appeal.

IV. **Status Of Amendments**

There have been no amendments filed subsequent to the rejection of July 27, 2007

V. Summary Of The Invention

Claims 1, 13, 18 & 26 are argued separately below. Claim 1 requires a template defining the second hierarchical data scheme. The template is shown in FIG. 1 as element 22, in FIG. 2 as element 44 and FIG. 11 shows an example of a template. According to the specification, page 5, lines 21-26, "The system 20 includes a template 22 that defines a second hierarchical data scheme. In one embodiment the second hierarchical data scheme may be a sample extensible markup language file, an XML target format template, a extensible markup language document type definition or an extensible markup language schema". Claim 1 further requires that the hierarchical data scheme is a scheme that groups data and its context. This is an inherent feature of XML (eXtensible Markup Language) and the other examples of hierarchical data schemes. The next element of claim 1 is a dynamic data generation module. The dynamic data generation module is shown in FIG. 1 as element 24, in FIG. 2 as element 46, 48 & 50. Claim 1 requires that the dynamic data generation module be contained in the template. According to the specification, page 6, lines 2-3 "A dynamic data generation module 24 is contained in the template 22." In addition the specification states, page 6, lines 5-12, "The dynamic data generation module 24 includes a query 28 to the data source 26 and driver for connecting to the data source 26. The results of the query, in one embodiment, place the acquired data in the appropriate part of the template. In another embodiment, the query takes the data from the appropriate part of the template and places it in the data store (26). In this way the data is converted from the first hierarchical data scheme to the second hierarchical data scheme or vice versa." The next element in claim 1 is a data source. A data source is shown in FIG. 1 as element 26 and in FIG. 2 as elements 56, 58. According to the specification page 6, lines 3-5 "A data source 26 is in communication with the dynamic data generation module 24. The data source 26 contains data in the first hierarchical data scheme" as required by claim 1.

Claim 13 requires publishing a dynamic template in a server. According to the specification page 7, lines 24-26, "A wizard 70 (FIG. 2) or set of wizards walk a user through the process of converting the static template 68 (FIG. 2) into a dynamic template 72 (FIG. 2) that is then published to a server 42(FIG. 2). The next step in claim

13 is receiving an instruction from a client at the dynamic template. According to the specification page 7, lines 11-12 "In one embodiment the client transmits a key and an instruction 62 to the server 42 and receives an XML document related to the key 62." The next step in claim 13 requires executing the dynamic template. This is shown as step 76 of FIG. 3. The next step in claim 13 requires when a dynamic data generation module is executed performing a data transfer operation that converts data in the first hierarchical data scheme into the second hierarchical data scheme. According to the specification pages 8-9, lines 25-26 & 1-2 "When a dynamic data generation module is executed at step 78, a data transfer operation is performed that converts data in the first hierarchical data scheme into the second hierarchical data scheme which ends the process at step 80." Claim 13 requires that the hierarchical data scheme is a scheme that groups data and its context which is inherent to XML and the other examples of hierarchical data schemes. According to the specification page 6, lines 12-21 "In one embodiment, the first hierarchical data scheme may be an extensible markup language scheme, a relational database, a non-relational database, an extensible markup language database, or a self describing database. The second hierarchical data scheme, in one embodiment, may be an extensible markup language scheme, a relational database, a non-relational database, an extensible markup language database, or a self describing database. Note that the first and second hierarchical data scheme may be in the same class (e.g., both relational databases) but have a different data scheme."

Claim 18 requires as its first step receiving a static extensible markup language template. This is shown as element 92 of FIG. 4 and discussed on page 10, lines 6-7. The next step of claim 18 is determining for each element of the static extensible markup language template if a datum needs to be dynamically generated. This is element 94 of FIG. 4 and discussed on page 10, lines 7-9. The next step of claim 18 is when the datum needs to be dynamically generated, receiving a data source having data in the hierarchical data scheme for acquiring the datum. This is step 96 of FIG. 4 and discussed on page 10, lines 9-11. The next step of claim 18 is receiving a data map between a data element in the data source and a metatag in the static extensible markup language template. This is step 98 of FIG. 4 and discussed on page 10, lines 11-13. The last step of claim 18 is repeating steps (b) through (d) for every element of

5

the static extensible markup language template to form a dynamic data conversion program. This is element 100 of FIG. 5 and discussed on page 10, lines 13-16.

Claim 26 requires as its first step receiving a sample extensible markup language file. This is element 112 of FIG. 6 and discussed on page 11, lines 12-14. The next step of claim 26 is determining for each element of the sample extensible markup language file if a datum needs to be dynamically processed. This is element 114 of FIG. 6 and discussed on page 11, lines 14-16. The next step of claim 26 is when the datum needs to be dynamically processed, receiving an extensible markup language element location for acquiring the datum. This is element 116 of FIG. 6 and discussed on page 11, lines 16-18. The next step of claim 26 is receiving a data map between a metatag in the sample extensible markup language file and an element of the hierarchical data scheme. This is element 118 of FIG. 6 and discussed on page 11, lines 18-20. The next step of claim 26 is repeating steps (b) through (d) for every element of the sample extensible markup file to form a dynamic data conversion program. This is element 120 of FIG. 7 and discussed on page 11, lines 20-23.

VI. Grounds of Rejection to be Reviewed on Appeal

1. Whether claims 7, 8, 9, 10, 11 & 12 are unclear under 35 USC 112, second paragraph?

2. Whether claims 1-12 & 18-29 are statutory under 35 USC 101?

3. Whether claims 1-3, 5-11 & 13-17 are anticipated under 35 USC 102(e) by Fernandez et al (US 6604100)?

4. Whether claims 4 & 12 are unpatentable under 35 USC 103(a) over Fernandez et al (US 6604100) in view of Prompt et al (US 6985905)?

5. Whether claims 18-23 & 25-29 are unpatentable under 35 USC 103(a) over Fernandez et al (US 6604100) in view of Krupa et al (US 20020156811)?

VII. Argument

This is the third Appeal Brief filed in this case, which was filed in 2001. MPEP 707.07(g) and 37 CFR1.104 require that the PTO provide all valid grounds for rejecting the claims in an application in a single action. This piecemeal examination is not allowed under the law, is expensive for both the applicant and the PTO and a waste of resources and time of both the PTO and the applicant. The applicant respectfully requests that a decision on the merits be entered in this case.

Issue 1   Whether claims 7, 8, 9, 10, 11 & 12 are unclear under 35 USC 112, second paragraph?

A claim is not indefinite if "one skilled in the art would understand all the language in the claims when read in light of the specification." *Rosemount, Inc. v. Beckman Instruments, Inc.*, 727 F.2d 1540, 221 USPQ 1, 7 (Fed. Cir. 1984). The Patent Office suggests that claim 7 is indefinite because of the word "if" (OA 7/27/07 page 2). The applicant disagrees that the word "if" makes the rest of the claim indefinite to one skilled in the art. However, the applicant is willing to change the word "if" to "when" and then the claim is clearly not indefinite.

Issue 2   **Whether claims 1-12 & 18-29 are statutory under 35 USC 101?**

This issue was raised in the last Appeal Brief. Clearly the Applicant prevailed on this issue. The Examiner should be precluded from re-raising this issue. MPEP 707.07(g) and 37 CFR1.104 require that the PTO provide all valid grounds for rejecting the claims in an application in a single action. This piecemeal examination is not allowed under the law, is expensive for both the applicant and the PTO and a waste of resources and time of both the PTO and the applicant.

"Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefore" 35 USC 101. The present invention is directed to a

problem that industry spends billions of dollars on each year trying to solve, namely allowing different computer systems to talk to each other. More specifically the invention is directed to converting the data in a first format into data in a second format in a computer system or systems. So clearly, the invention is useful. The invention is directed to a process and machine. The invention converts data in a first hierarchical data scheme into a second hierarchical data scheme. This is clearly a process by any definition of the word process. The invention is also a machine. It is an electronic circuit that just happens to be a general purpose computer or computers that are configured into a specific electronic circuit using software to solve a specific problem.

The Examiner (OA 7/27/07 page 3) suggests the claim 1 is directed to software per se. This is clearly incorrect, since the claim recites a data source which is a piece of computer hardware. In addition, the claim states that the data source is in communication which means there is a communication system. Claims 1-12 are clearly statutory.

With respect to claims 18 & 26 the Examiner (OA 7/27/07 page 3) suggests that there is no tangible result. The investors in the present assignee and the customers of the assignee of the present invention would be amazed that they had paid good money to obtain a "non-real world" value (OA 7/27/07). The present invention clearly has a tangible result, whether looked at from an economic point of view or from a physical point of view. From an economic point of view the present invention is directed to a portion of a problem that commercial companies have been trying to solve since the first computers were deployed. From a physical point of view a general purpose computer running software is a specific machine or electronic circuit. Electronic circuits are clearly patentable and the present invention is an electronic circuit. The computer running the specific software causes switches to open and close and electronic charges to be stored. This is clearly a physical process and therefore a machine.

The rejection of claims 18-29 under 35 USC 101 has no support in law or logic and must be withdrawn.

Issue 3.   Whether claims 1-3, 5-11 & 13-17 are anticipated under 35 USC 102(e) by Fernandez et al (US 6604100)?

A prior art reference must inherently or expressly disclose every element and the elements must be arranged as in the claim for a proper anticipation rejection under section 102. *Richardson v. Suzuki Motor Co.*, 868 F.2d.1226, 9 USPQ2d 1913 (Fed. Cir. 1989)

Claim 1 recites a dynamic data generation module contained in the template. The Patent Office states (OA 7/27/07 page 5) that the template of Fernandez is the "XML construction part, col. 5, lines 29 Fernandez). The Patent Office states (OA 7/27/07 page 5) that the dynamic data generation module is the XML module generator (106, FIG. 1). Fernandez (Col. 5, lines 26-29) clearly states that the translator 104 partitions the executable query "into one or more SQL queries and an XML template". According to Fernandez (col. 9, lines 62-64 & col. 5, lines 43-45) the SQL queries are executed and their results are merged into the XML-construction part (XML template) by the XML module generator 106. Clearly the XML module generator 106 is not contained in the template (XML-construction part) since the generator 106 uses the template (XML-construction part) in conjunction with the results of one or more SQL queries. Fernandez clearly does not inherently or expressly disclose every element and the elements arrangement as in the claim. Claim 1 is clearly allowable.

Claims 2-3 are allowable for the same reasons as claim 1.

Claim 5 recites that the template is a static extensible markup language document. The Patent Office points to col. 5, lines 43-45 (OA dated 10/2/06, page 6). However, a close reading of this section shows that it just discusses merging the tuple streams with the XML-construction part. This does not tell whether the XML-construction part is a static XML document. Fernandez clearly does not inherently or expressly disclose every element and the elements arrangement as in the claim. Claim 5 is clearly allowable.

Claims 6 & 7 are allowable for the same reasons as claim 5.

Claims 7-9 are allowable as being dependent upon an allowable base claim.

Claim 10 recites that the dynamic data generation module includes a query. The Patent Office states (OA 7/27/07 page 5) that the dynamic data generation

10

module is the XML module generator (106, FIG. 1). However, element 106 does not contain any queries since, according to Fernadez (col. 9, lines 62-64 & col. 5, lines 43-45) the SQL queries are executed and their results are merged into the XML-construction part (XML template) by the XML module generator 106. Thus the SQL queries are executed outside the XML module generator (106, FIG. 1). Thus Fernadez does not show the dynamic data generation module includes a query. Claim 10 is clearly allowable.

Claim 11 recites that the dynamic data generation modules includes a mapping between the first hierarchical data scheme and the second hierarchical data scheme. Note that the Patent Office has stated that the dynamic data generation module of is the XML module generator (106, FIG. 1). The Patent Office (OA 7/27/07 page 7) points to Col. 2, lines 42-51 of Fernandez. However, this section clearly states that the mapping is done by a new language RXL. RXL is not part of the XML module generator (106, FIG. 1), see Col. 5, lines 15-25 which states the RXL is part of the composer module 102. Clearly, Fernandez does not inherently or expressly disclose every element and the elements must be arranged as in the claim. Claim 11 is allowable.

Claims 13-17

Claim 13 requires a dynamic template. The Patent Office points to the query composer 102 of Fernandez. However, the query composer 102 is not a template, which is defined as a pattern in the present application. The only template mentioned in Fernandez is the XML-construction part e.g. XML template, Fernandez Col. 5, lines 27-29. There is no suggestion that the XML-construction part is executable in Fernandez. Clearly, Fernandez does not inherently or expressly disclose every element and the elements must be arranged as in the claim. Claim 13 is allowable.

Claim 14 recites determining for each element in the template if dynamically generated data is required. There is just no suggestion or teaching in Fernandez that the XML-construction part is executable. Clearly, Fernandez does not inherently or expressly disclose every element and the elements as arranged as in the claim. Claim 14 is allowable.

11

Claims 15-17 are allowable as being dependent upon an allowable base claim.

Issue 4.   Whether claims 4 & 12 are unpatentable under 35 USC 103(a) over Fernandez et al (US 6604100) in view of Prompt et al (US 6985905)?

The question of obviousness requires that we determine if the references, taken as a whole, would suggest the invention to one of ordinary skill in the art. *Medtronic, Inc. v. Cardiac Pacemakers, Inc.*, 721 F.2d 1563, 220 USPQ 97 (Fed. Cir. 1983).

Claims 4 & 12 are allowable for the same reasons as claim 1 above. The addition of Prompt et al does not resolve this issue.

Issue 5.   Whether claims 18-23 & 25-29 are unpatentable under 35 USC 103(a) over Fernandez et al (US 6604100) in view of Krupa et al (US 20020156811)?

The question of obviousness requires that we determine if the references, taken as a whole, would suggest the invention to one of ordinary skill in the art. *Medtronic, Inc. v. Cardiac Pacemakers, Inc.*, 721 F.2d 1563, 220 USPQ 97 (Fed. Cir. 1983).

Claim 18 recites determining for each element if a datum needs to be dynamically generated. The Patent Office points to the query composer 102 of Fernandez. However, the query composer 102 is not a template, which is defined as a pattern in the present application. The only template mentioned in Fernandez is the XML-construction part e.g. XML template, Fernandez Col. 5, lines 27-29. There is no suggestion that the XML-construction part is executable in Fernandez. The addition of Krupa does not solve this and the Patent Office cites Krupa for repeating the steps. The combination of Fernandez and Krupa taken as a whole, would not suggest the invention to one of ordinary skill in the art. Claim 18 is clearly allowable.

Claims 19-23 are allowable as being dependent upon an allowable base claim.

Claim 25 is allowable for the same reasons as claim 18

Claim 26 recites determining for each element if a datum needs to be dynamically generated. The Patent Office points to the query composer 102 of Fernandez. However, the query composer 102 is not a template, which is defined as a pattern in the present application. The only template mentioned in Fernandez is the XML-construction part e.g. XML template, Fernandez Col. 5, lines 27-29. There is no suggestion that the XML-construction part is executable in Fernandez. The addition of Krupa does not solve this and the Patent Office cites Krupa for repeating the steps. The combination of Fernandez and Krupa taken as a whole, would not suggest the invention to one of ordinary skill in the art. Claim 26 is clearly allowable.

Claims 27-29 are allowable as being dependent upon an allowable base claim.

IX. **Appendix Of The Appealed Claims**

1.   A system for converting data in a first hierarchical data scheme into a second hierarchical data scheme, comprising:

a template defining the second hierarchical data scheme, wherein a hierarchical data scheme is a scheme that groups data and its context;

a dynamic data generation module contained in the template; and

a data source, in communication with the dynamic data generation module, containing data in the first hierarchical data scheme.

2.   The system of claim 1, wherein the template and the dynamic data generation module are contained in a server.

3.   The system of claim 2, further including a driver connected between the dynamic data generation module and the data source.

4.   The system of claim 3, further including a developer module contained in the server for creating the dynamic data generation module.

5.   The system of claim 1, wherein the template is a static extensible markup language document.

6.   The system of claim 1, wherein the template is an extensible markup language document type definition.

7.   The system of claim 1, wherein the template is an extensible markup language schema.

8.   The system of claim 1, wherein the first hierarchical data scheme is selected from the group of: extensible markup language schemes, relational databases, non-relational databases, extensible markup language databases and self describing databases.

9.   The system of claim 1, wherein the second hierarchical data scheme is selected from the group of: extensible markup language schemes, relational databases, non-relational databases, extensible markup language databases and self describing databases.

10.   The system of claim 1, wherein the dynamic data generation module includes a query directed to the data source.

11.   The system of claim 1, wherein the dynamic data generation module includes a data mapping between the first hierarchical data scheme and the second hierarchical data scheme.

12.   The system of claim 4, wherein the developer module contains a wizard that walks a user through a process of creating the dynamic data generation module.

13.   A method of converting data in a first hierarchical data scheme into a second hierarchical data scheme, comprising the steps of:

a) publishing a dynamic template in a server;
b) receiving an instruction from a client at the dynamic template;
c) executing the dynamic template; and
d) when a dynamic data generation module is executed, performing a data transfer operation that converts data in the first hierarchical data scheme into the second hierarchical data scheme, wherein a hierarchical data scheme is a scheme that groups data and its context.

14. The method of claim 13, wherein step (a) further includes the steps of:

a1) receiving a template;

a2) determining for each element of the template if a dynamically generated data is required;

a3) when the dynamically generated data is required, receiving a data source for obtaining the dynamically generated data.

15. The method of claim 14, further including the steps of:

a4) receiving a data mapping between the first hierarchical data scheme and the second hierarchical data scheme.

16. The method of claim 15 wherein step (a4) further includes the steps of:

i) when the first hierarchical data scheme is a non-extensible markup language and the second hierarchical data scheme is a second non-extensible markup language, creating a first data mapping between the first hierarchical data scheme and an intermediate extensible markup scheme;

ii) creating a second data mapping between the intermediate extensible markup scheme and the second hierarchical data scheme.

17. The method of claim 15, further including the step of"

a5) receiving a key associated with the data mapping.

18.   A method of converting data in a hierarchical data scheme into an extensible markup language scheme, comprising the steps of:

a) receiving a static extensible markup language template;

b) determining for each element of the static extensible markup language template if a datum needs to be dynamically generated;

c) when the datum needs to be dynamically generated, receiving a data source having data in the hierarchical data scheme for acquiring the datum;

d) receiving a data map between a data element in the data source and a metatag in the static extensible markup language template; and

e) repeating steps (b) through (d) for every element of the static extensible markup language template to form a dynamic data conversion program.

19.   The method of claim 18, wherein step (a) further includes the step of receiving a template selected from the group including: an extensible markup language document type definition and an extensible markup language schema.

20.   The method of claim 18, wherein step (a) further includes the step of:

a1) defining an input parameter.

21.   The method of claim 18, wherein step (c) further includes the step of:

c1) receiving a driver.

22.   The method of claim 18, wherein step (c) further includes the step of:

c1) generating a query to the data source.

23. The method of claim 18, wherein step (d) further includes the step of:

   d1) receiving a screen having a list of elements from the data source and a list of metatags from the static extensible markup language template.

24. The method of claim 18, wherein step (c) further includes the step of:

   c1) displaying an incomplete version of a dynamic extensible markup language template wherein a static element is shown in a first color and a dynamic element is shown in a second color.

25. The method of claim 18, further including the steps of:

   e) publishing the dynamic data conversion program to a server;
   f) when a query is received at the server for the dynamic data conversion program, executing the dynamic data conversion program to form an extensible markup language document.

26. A method of converting data in an extensible markup language scheme into a hierarchical data scheme, comprising the steps of:

   a) receiving a sample extensible markup language file;
   b) determining for each element of the sample extensible markup language file if a datum needs to be dynamically processed;
   c) when the datum needs to be dynamically processed, receiving an extensible markup language element location for acquiring the datum;
   d) receiving a data map between a metatag in the sample extensible markup language file and an element of the hierarchical data scheme; and
   e) repeating steps (b) through (d) for every element of the sample extensible markup file to form a dynamic data conversion program.

27. The method of claim 26, wherein step (a) further includes the step of:

a1) defining a key.

28. The method of claim 26, wherein step (d) further includes the steps of:

d1) receiving a query type;
d2) generating a query.

29. The method of claim 28, wherein step (d1) further includes receiving an insert query type.
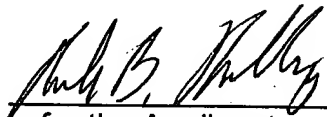
IX. Evidence Appendix

None

X. Related Proceedings Appendix

None

Respectfully submitted,
(Vandersluis)

By: _____
Attorney for the Applicant
Dale B. Halling
Registration No. 38,170
Customer No. 25,007
Phone: (719) 447-1990
Fax: (719) 447-9815